

FLOATING POINT ARITHMETIC PROCESSOR

Patent Number: JP9258959
Publication date: 1997-10-03
Inventor(s): NAKAZATO SATOSHI
Applicant(s):: NEC CORP
Requested Patent: JP9258959
Application Number: JP19960067677 19960325
Priority Number(s):
IPC Classification: G06F7/00
EC Classification:
Equivalents: JP2868075B2

Abstract

PROBLEM TO BE SOLVED: To reduce the hardware quantity of a floating point computing element dealing with the floating point of an IEEE-754 system and to shorten the processing time of the floating point computing element.

SOLUTION: A part for judging whether read data applies to any special number and outputting a flag signal 668 showing the type and a part for rewriting rewritten data when rewritten data applies to the special number by a following flag signal 668 are added to a storage device at the outer side of a floating point arithmetic processor and a LOAD/STORE unit 660 for reading/ writing data. Routes 611, 612 and 631 for transferring the flag signals with numeric data are provided in the floating point arithmetic processor. Thus, only simple logic circuits 655 generating the flag signals 656 and 657 of input data and the flag signal 658 of output data from output data 653 of a computing element 650 are provided for the respective floating point computing elements 650.

Data supplied from the esp@cenet database - I2

【特許請求の範囲】

【請求項1】 少なくとも1つ以上の浮動小数点演算器と、前記浮動小数点演算器への入力データと前記浮動小数点演算器からの出力データとを記憶する記憶装置と、少なくとも1つ以上の前記記憶装置から前記浮動小数点演算器への入力データを読み出す手段と、前記読み出し手段と前記浮動小数点演算器との間で前記読み出しデータを転送する1つ以上の経路と、少なくとも1つ以上の前記浮動小数点演算器から前記記憶装置へ出力データを書き戻す手段と、前記浮動小数点演算器と前記書き戻し手段との間で書き戻しデータを転送する1つ以上の経路と、を備える処理装置において、前記記憶装置から前記浮動小数点演算器へのデータを読み出す手段において読み出しデータが予め定められた浮動小数点形式の特定のデータの種類の一致することを検出しその状態を前記読み出しデータとは別のデータとして出力する手段と、前記特定のデータの種類の表すデータを入力とする演算結果を同様の特定のデータの種類の表すデータで出力する手段と、前記浮動小数点演算器の出力データを前記記憶装置に書き戻す手段において前記特定のデータの種類の表すデータを出力する手段より得られるデータに応じて前記浮動小数点演算器の出力データを前記特定のデータに置き換える手段とを備えることを特徴とする浮動小数点演算処理装置。

【請求項2】 前記浮動小数点演算器は複数の入力データを選択する手段を有し、演算結果の出力データを他の浮動小数点演算器の入力へ転送する経路と、前記出力データの特定の種類の表す出力データを他の浮動小数点演算器の入力へ転送する経路とを有し、前記複数の浮動小数点演算器が前記経路によって相互に接続されたことを特徴とする請求項1記載の浮動小数点演算処理装置。

【請求項3】 前記データ読み出し手段からのデータと、前記1つ以上の浮動小数点演算器からのデータと、を一時的に記憶する記憶装置と、前記読み出しデータの特定のデータの種類の表すデータと、前記書き戻しデータの特定のデータの種類の表すデータと、を一時的に記憶する記憶装置とを備えることを特徴とする請求項1記載の浮動小数点演算処理装置。

【請求項4】 前記浮動小数点演算器において、入力データの特定の種類の表すデータが有効なデータであった場合に通常の入力データによる演算の実行を停止する手段を有することを特徴とする請求項1記載の浮動小数点演算処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、浮動小数点演算器に関し、特にIEEE-754に規定される形式の数値データの演算を行う浮動小数点演算処理装置に関する。

【0002】

【従来の技術】 浮動小数点のデータ形式として「IEEE

Standard for Binary Floating Point Arithmetic, ANSI/IEEE std 754-1985」(以下、IEEE-754形式)が事実上標準として使用されるようになってきた。IEEE-754形式は浮動小数点形式の数値データを指数部と仮数部とに分け、これらの数値の組み合わせにより通常数と、5種類の特殊数とを定義している。

【0003】 浮動小数点演算器でIEEE-754形式を扱う場合、通常数のみが必ず入力されるようなフォーマットに対応する演算器と異なり特殊数に対する特別な対応が必要になる。すなわち、演算器に入力される複数の数値データがそれぞれ通常数であるかまたはいずれかの特殊数であるかを判別し、その結果どれか1つでも入力数値データが特殊数に該当したり、また通常数同士の演算であってもオーバーフローやアンダーフローのような例外が発生する場合にはその結果に応じて演算器の出力結果の数値データを正しい特殊数の数値データに置き換える必要がある。また、「特開平4-281518 デジタルシグナルプロセッサ」においては従来IEEE-754形式を扱っていなかったデジタルシグナルプロセッサにおいて、簡単なハードウェアによって外部メモリ中に配置されたIEEE-754形式の数値データを扱う方法が提示されている。

【0004】 しかし、これはデジタルシグナルプロセッサ内部の数値データ形式への変換する方法を示したものである。この中ではIEEE-754形式の特殊数をデジタルシグナルプロセッサ内部で用いられる2の補数形式へと変換するハードウェア構成について述べられているが、演算器自体がこれら特殊数をどのように扱うかについてまでは触れられていない。

【0005】

【発明が解決しようとする課題】 第1の問題点は、このようなIEEE-754形式を扱う浮動小数点演算器では演算器全体のハードウェア量が増大することである。

【0006】 その理由は、入力される複数の数値データ毎にその数値データがIEEE-754形式で定義されるどの数値データの種類の該当するかの判別するハードウェアと、ここで得られた数値データの種類の、たとえばそれが通常数同士の演算であってもオーバーフローやアンダーフローの例外発生を起こす場合には演算器からの出力結果の数値データを適切な特殊数の数値データで置き換えるためのハードウェアが必要になるからである。最近ではLSIの集積度が非常に高くなり、マイクロプロセッサもスーパースカラやVLWなどのアーキテクチャをとると1チップに複数の浮動小数点演算器を搭載することも珍しくなり、ハードウェア量増大の影響は非常に深刻である。

【0007】 第2の問題点は、演算器全体の処理時間が増えることにより性能が低下することである。

【0008】その理由は、前記の第1の問題点の理由で示した新たなハードウェアが必要になることから通常の演算器での処理に加えてこれら追加ハードウェアでの処理が必要になるからである。特に出力結果の数値データを置き換える処理は演算器内部の他の処理と並列に実行することができないため、この部分での処理時間がそのまま増加することになる。クロックサイクルの高速化を行うに際し、このような演算器の処理時間増大は大きな影響を与えることは明かである。

【0009】本発明では、前記のような問題点を解決し、IEEE-754形式の数値データを扱う浮動小数点演算処理装置で、特に特殊数を扱う際のハードウェア量削減並びに演算処理時間の短縮が可能な浮動小数点演算処理装置を提供することを目的とする。

【0010】

【課題を解決するための手段】本発明における浮動小数点演算処理装置は、少なくとも1つ以上の浮動小数点演算器と、前記浮動小数点演算器への入力データと前記浮動小数点演算器からの出力データとを記憶する記憶装置と、少なくとも1つ以上の前記記憶装置から前記浮動小数点演算器への入力データを読み出す手段と、前記読み出し手段と前記浮動小数点演算器との間で前記読み出しデータを転送する1つ以上の経路と、少なくとも1つ以上の前記浮動小数点演算器から前記記憶装置へ出力データを書き戻す手段と、前記浮動小数点演算器と前記書き戻し手段との間で書き戻しデータを転送する1つ以上の経路とを備え、前記記憶装置から前記浮動小数点演算器へのデータを読み出す手段において読み出しデータが予め定められた浮動小数点形式の特定のデータの種類の一致することを検出しその状態を前記読み出しデータとは別のデータとして出力する手段と、前記特定のデータの種類の表すデータを入力とする演算結果を同様の特定のデータの種類の表すデータで出力する手段と、前記浮動小数点演算器の出力データを前記記憶装置に書き戻す手段において前記特定のデータの種類の表すデータを出力する手段より得られるデータに応じて前記浮動小数点演算器の出力データを前記特定のデータに置き換える手段とを新たに加えることにより構成される。

【0011】

【作用】浮動小数点演算器に入力される数値データは、本発明の浮動小数点演算処理装置の外部の記憶装置からデータを読み出し演算器へと供給する手段においてIEEE-754形式の通常数かそれとも特定の特殊数に該当するかを同時に検出してこれらの特定の種類の表すフラグ信号を数値データに付加する。

【0012】また、本発明の浮動小数点演算処理装置から外部の記憶装置に数値データを書き戻す手段においてIEEE-754形式の特定の特殊数を表すフラグ信号から実際に書き戻す数値データを正しいデータパターンに置き換える。

【0013】本発明の浮動小数点演算装置内の各浮動小数点演算器は入力される数値データとそれに付随するフラグ信号とを見て、フラグ信号が特殊数を表していた場合にはこの入力フラグ信号から組み合わせ論理回路により出力結果の数値データの特殊数を表すフラグ信号を生成する。この際に演算器から実際に出力される数値データは無視することにする。本発明の浮動小数点演算装置内ではこのフラグ信号を常に数値データとともに転送することで、各演算器において特殊数を検出するハードウェアや出力を置き換えるハードウェアを不要にすることができる。

【0014】

【発明の実施の形態】次に、本発明の実施形態について図面を参照して説明する。

【0015】図1はIEEE-754形式の倍精度におけるフォーマットとその際の特殊数の種類の表すデータパターンを表した図である。

【0016】IEEE-754形式の倍精度では、符号1ビット、指数11ビット、仮数52ビットからなる。そして指数と仮数の特定のデータパターンにより零(ZERO)、無限大(∞)、2種類の非数(sNaN、qNaN)、微小固定小数点数(denormal)の計5種類の特殊数が規定される。なお、これらの特殊数以外の数を通常数(normal)と呼ぶ。

【0017】図2は、このようなIEEE-754形式を扱う従来の浮動小数点演算器の例を示すブロック図である。

【0018】2つの入力オペランドデータ200、201を入力とする浮動小数点演算器(演算器本体)240以外に、それぞれの入力オペランドデータ200、201が特殊数のいずれかのデータパターンに一致するかどうかを検出し、それを適当な形式のフラグ信号(以下、オペランドステータスとする)210、211を生成するオペランドステータス生成回路251、252がある。浮動小数点演算器(演算器本体)240の出力はアンダーフローやオーバーフロー時の例外処理を行う例外処理回路260へ送られる。一方、オペランドステータス210、211はその状態に応じて出力データのオペランドステータス230を決定するための簡単な組み合わせ論理回路である出力ステータス生成回路270へ送られる。例外処理回路260を通った出力データ220は、この出力データ220のオペランドステータス230が特殊数を示していた場合にはその種類に応じたデータパターンに置き換えられる必要がある。これを行うのが出力差し替え回路280である。

【0019】図3はこのような浮動小数点演算器を構成するオペランドステータス生成回路251又は252の詳細を説明するブロック図である。

【0020】オペランドステータス生成回路251又は252は、入力オペランドデータ300が特殊数のデー

タパターン301, 302, 303, 304, 305と一致するかどうかを検出する複数の比較器311, 312, 313, 314, 315と、それらの比較器311, 312, 313, 314, 315の出力から所望のオペランド・ステータス330を生成するためのエンコーダ320とから成っている。

【0021】また、図4は図2に示すような浮動小数点演算器を構成する出力差し替え回路280の詳細を説明するブロック図である。

【0022】出力差し替え回路280は、演算器からの通常数の場合の出力データ400と各特殊数のデータパターン401, 402, 403, 404, 405を入力とする選択回路430と、差し替えられる出力データのオペランド・ステータス410から選択回路430の切り替え制御信号を生成するデコーダ420とから成っている。

【0023】以上のように、IEEE-754形式を扱う従来の浮動小数点演算器は各演算器毎に入力オペランドが特殊数であるかどうかを検出してはそれに応じて出力データを差し替えるという構成を採っていた。

【0024】本発明における浮動小数点演算処理装置ではオペランドと同時にオペランドステータスを装置内で持ち回ることにより特殊数の場合の処理の簡単化を行う。

【0025】図5が、本発明の実施形態に係る浮動小数点演算処理装置内の各浮動小数点演算器の構成を示すブロック図である。

【0026】複数の入力オペランド500, 501により演算を行い出力オペランド520を生成する演算器本体540と、各入力オペランド500, 501に付随する入力オペランド・ステータス510, 511と演算器本体540の出力オペランド520を入力として出力オペランド520のオペランド・ステータスを生成するための組み合わせ論理回路である出力オペランドステータス生成回路570とから構成される。

【0027】入力オペランドステータスの少なくとも一方が特殊数のステータスを表していた場合には、出力オペランドステータスは予め規定されているルールに従っていずれかの特殊数のステータスを出力する。また、入力オペランドステータスがいずれも通常数である場合でも、演算器本体540の出力オペランド520によってはアンダーフローやオーバーフローが発生することがある。IEEE-754ではアンダーフロー時の出力はZEROとなるよう規定されており、またオーバーフロー時には丸めモードに応じて ∞ か通常数で採り得る最大の値のいずれかを出力するように規定されている。従って、出力オペランドステータス生成回路570は出力オペランド520も入力として前記のような例外が発生した場合には適切な出力オペランドステータスを出力するように論理回路を構成する。

【0028】なお、オーバーフロー時に出力される可能性のある通常数の採り得る値の中で最も大きな値（以下、最大数）は本来特殊数ではないが例外時に出力されること、及び特定のデータパターンになることを考慮して、5種類の特殊数のオペランドステータス以外に通常数以外のオペランドステータスとして追加することが充分理にかなっていることは明かである。

【0029】図6は、本発明における浮動小数点演算処理装置全体の一実施形態を表すブロック図である。

【0030】特に、整数演算処理系と浮動小数点演算処理系とが分離したマイクロプロセッサの浮動小数点演算処理系に応用した例である。

【0031】データ読み出しポート641, 642とデータ書き込みポート643を有する浮動小数点レジスタファイル(RF)640と、ステータス読み出しポート646, 647とステータス書き込みポート648を有する浮動小数点ステータスレジスタファイル(RF)645と、浮動小数点演算器650と出力オペランドステータス生成回路655と、図面外の記憶装置との間でデータの入出力を行うLOAD/STOREユニット660とで構成され、それぞれの構成単位はオペランドデータバス601, 602, 621とオペランドステータスバス611, 612, 631とで相互に接続されている。

【0032】本実施形態に係る浮動小数点演算処理装置の動作を図を用いて説明する。

【0033】浮動小数点レジスタファイル(RF)640には複数の浮動小数点データが蓄えられている。一方、各々の浮動小数点データに対するオペランドステータスが、浮動小数点ステータスレジスタファイル(RF)645の中に浮動小数点レジスタファイル(RF)640のアドレスと同一のアドレスに蓄えられている。浮動小数点レジスタファイル(RF)640中から必要なデータを、図面外の命令処理系から与えられるアドレスにより読み出しポート641, 642に出力する。この際、同一のアドレスが浮動小数点ステータスレジスタファイル(RF)645にも与えられ、各々のデータに対応したオペランドステータスが読み出しポート646, 647に出力される。オペランドデータは浮動小数点レジスタファイル640の読み出しポート641, 642よりオペランドデータバス601, 602に出力され、これらのオペランドデータを必要とする他の処理要素へと転送される。同時にオペランドステータスも浮動小数点ステータスレジスタファイル645の読み出しポート646, 647よりオペランドステータスバス611, 612に出力され転送される。

【0034】浮動小数点演算器650は、オペランドデータバス601, 602より必要なオペランド651, 652を獲得し、演算結果653を出力する。これと同時に出力オペランドステータス生成回路655は浮動小

数点演算器650への入力オペランド651、652に付随してオペランドステータスバス611、612上にある各々のオペランドステータス656、657を獲得し、浮動小数点演算器650の出力オペランド653の例外検出も行いながら、この出力オペランド653に付随すべきオペランドステータス658を生成する。出力オペランド653と出力されたオペランドステータス658は、それぞれオペランドデータバス621、オペランドステータスバス631に出力され他の処理要素へと転送される。

【0035】通常は再びレジスタファイルへと書き戻される場合が多く、オペランドデータは浮動小数点レジスタファイル(RF)640の書き込みポート643から書き戻され、オペランドステータスは書き戻された浮動小数点レジスタファイル(RF)640のアドレスと同一の浮動小数点ステータスレジスタファイル(RF)645中のアドレスに書き込みポート648から書き戻される。

【0036】このように、本実施形態の浮動小数点演算処理装置内では、オペランドデータにそのデータの種別を示すオペランドステータスを付随して転送することで各演算器が持っていた機能を省いている。しかし、本実施形態の浮動小数点演算処理装置外とデータのアクセスを行う場合には、本来のIEEE-754形式に準拠したデータとする必要がある。この役割をLOAD/STOREユニット660が担当する。

【0037】図7は、LOAD/STOREユニット660(図6参照)の詳細を表すブロック図である。

【0038】LOAD/STOREユニット660(図6参照)は、外部の記憶装置にアクセスするための実行アドレスを生成するためのアドレス加算器750と論理アドレスから物理アドレスへの変換を行うテーブルであるTLB(Translation Lookaside Buffer)760と、書き戻しデータをIEEE-754形式に準拠させるためのオペランド差し替え回路770と、読み込みデータから本実施形態の浮動小数点演算処理装置内で使用するためのオペランドステータスを生成するステータス生成回路780と外部メモリに対して書き込みデータを出力するのか、それとも読み出しデータを入力するのかメモリアクセスバス740を切り替えるためのスイッチ790とから成る。

【0039】次に、このLOAD/STOREユニット660(図6参照)の動作を説明する。

【0040】本実施形態の浮動小数点演算処理装置外の整数演算処理系より実行アドレス算出のためのオペランドデータ700、701が入力される。アドレス加算器750にてこれらのオペランドデータ700、701の加算を行い実行論理アドレスを得る。実際の外部メモリにアクセスするためには論理アドレスを物理アドレスに変換する必要がある。このためアドレス加算器750に

出力の一部を用いてTLB760をアクセスして物理アドレス730を得る。なお、本実施形態のマイクロプロセッサではこのような論理アドレスから物理アドレスへの変換を必要としているためにこのような構成を採用しているが、例えばデジタルシグナルプロセッサのように論理アドレスを持たない場合には、本実施形態に記載のTLB760は必要ないことは明かである。

【0041】一方、データ系の処理としてデータ書き戻しの際の動作をまず説明する。

【0042】通常書き戻しデータはそのままメモリアクセスバス740に出力すればよいが、本実施形態の浮動小数点演算処理装置ではIEEE-754形式の特殊数をオペランドステータスとして表し、実際のデータパターンには正しい値が設定されていない。従ってこのままデータだけを書き戻すことはできないため、オペランドステータス711の値によって書き戻しデータ710のデータパターンを適切なものに差し替えるオペランド差し替え回路770が必要である。このオペランド差し替え回路770は、従来の各浮動小数点演算器の出力部の存在した回路とほぼ同一の構成を採ることで実現可能であり、例えば図4にて示した構成などが利用できる。

【0043】次に、データ読み出しの際の動作を説明する。

【0044】読み出しデータも通常数の場合には特に問題にならないが、特殊数のデータを読み出した場合、本発明の浮動小数点演算処理装置内ではオペランドステータスが存在しないと特殊数を認識することができない。従って、読み出し(Read)データ720のデータパターンを調べて特殊数であった場合には、同時に適切なオペランドステータス721を出力するステータス生成回路780が必要になる。このステータス生成回路780も従来の浮動小数点演算器の入力部に存在した回路とほぼ同一の構成を採ることで実現可能であり、例えば図3にて示した構成などが利用できる。

【0045】図6に示した本発明の一実施形態に戻って説明をすると、LOAD/STOREユニット660より外部メモリにデータを書き戻す場合には、オペランドデータバス601より書き戻しデータ661を、オペランドステータスバス611よりそのデータに付随するオペランドステータス666をLOAD/STOREユニット660へ転送し、前記の動作に従ってオペランドの差し替えを行った後メモリアクセスバス680を用いて外部メモリへ書き戻される。LOAD/STOREユニット660へ入力されるオペランドデータ661とオペランドステータス666は直接浮動小数点レジスタファイル(RF)640と浮動小数点ステータスレジスタファイル(RF)645から転送されてくる場合が多い。

【0046】一方、外部メモリよりデータを読み出す場合には、メモリアクセスバス680より入力されたデータをLOAD/STOREユニット660内にて前記の

動作に従ってオペランドデータ663とそれに付随するオペランドステータス668を生成する。そして、それぞれオペランドデータバス621とオペランドステータスバス631により他の構成要素に転送される。これらのデータとステータスはそのまま浮動小数点レジスタファイル(RF)640と、浮動小数点レジスタファイルに書き込んだアドレスと同一の浮動小数点ステータスレジスタファイル(RF)645のアドレスに書き込まれる場合が多い。

【0047】このように本実施形態では、従来各演算器毎に個別に行っていた特殊数に対する処理を、各オペランドデータにオペランドステータス信号を付加することで不要にし、本実施形態の浮動小数点演算処理装置外とデータのアクセスを行う部分に前記の特殊数に対する処理を一括化することでハードウェアの削減、処理の簡単化を行っている。

【0048】次に、前記の一実施形態を一般化した例を図8に示す。

【0049】演算器を n 個850-1, ..., 850- n 、各演算器の出力オペランドステータス決定論理回路855-1, ..., 855- n 、LOAD/STOREユニットを m 個860-1, ..., 860- m とする。また、オペランドデータバスを多重化801-1, ..., 801- u , 802-1, ..., 802- u , 821-1, ..., 821- v とし、各々のオペランドステータスバス811-1, ..., 811- u , 812-1, ..., 812- u , 831-1, ..., 831- v もオペランドデータバスに合わせて多重化する。当然のことながら、同時に複数のデータを多重化バスにのせて各処理要素間に転送するために浮動小数点レジスタファイル(RF)840と浮動小数点ステータスレジスタファイル(RF)845もマルチポート化されている。

【0050】基本的な動作は、図6にて説明した動作と同一であり、同時に異なる複数のデータが転送され、それぞれ異なる演算処理が複数の処理要素にて実行されることになる。

【0051】なお、図8において各処理要素であるレジスタファイルと演算器、LOAD/STOREユニット間を多重化バス構成で接続しているが、もちろんこの接続方法だけに限られることはなく、レジスタファイルの各ポートと特定の処理要素を直接接続する構成や、逆に自由に接続先を変更できるクロスバー構成とするなどの変形が可能なのは明かである。

【0052】このような変形例の一例を図9に示す。

【0053】演算器#1(950-1)と演算器#2(950-2)の2つの演算器を持つ。出力オペランドステータス決定論理回路も各演算器対応で2つ955-1, 955-2存在する。各演算器の入力オペランド部にはオペランドデータバスからのオペランドと他方の演算器の出力オペランドとの入力を切り替える選択回路9

91-1, 992-1, 及び991-2, 992-2が存在する。対応する出力オペランドステータス決定回路955-1, 955-2の入力部にも同様な選択回路996-1, 997-1, 及び996-2, 997-2が存在する。

【0054】一方の演算器の出力を他方の演算器ですぐに使用したい場合に、一度バス経由でレジスタファイルに書き戻すのではなく、選択回路を切り替えて他方の演算器の出力オペランドを直接入力して処理時間の短縮をはかる構成である。このような場合にも、オペランドデータと同時にオペランドステータスも直接入力するバスとそのための選択回路を設けることで本発明を適用することが可能である。

【0055】なお、実施例においてレジスタファイルを有する構成を主に説明したが、本発明においてレジスタファイルの存在は必須ではないのは明かである。外部とのアクセスを実行する処理要素と実際の演算を行う処理要素とがあれば本発明の適用は可能である。

【0056】また逆に、本実施形態の浮動小数点演算処理装置内に含まれる記憶装置としてはレジスタファイルのみに限られるものではないことも明かである。特にアクセス速度を速くする必要がある同一マイクロプロセッサ内に集積される一次データキャッシュなどは、本発明の実施例によるLOAD/STOREユニットにおける付加回路による若干の遅延時間増が問題になるのであれば、一次データキャッシュにもステータス信号を同時に記録する領域を設けることで本発明の適用が可能になる。

【0057】また、上記実施形態では各演算器への入力オペランドのいずれか1つでも特殊数を表すステータスを持っていたならば、事実上ステータス信号のみが有効であり演算器への実際の入力オペランドデータ、及びこれらの演算結果である出力オペランドデータは全て無効なデータとして無視されることになる。従って、無効なデータに対する無駄な演算を行わないようにしようとする構成も可能で、図示はしていないが例えば、出力オペランドステータス決定論理回路より入力の方が特殊数であった場合に演算器本来へのクロック供給を止めたり、演算供給本体内の全F/Fをホールド状態にするなどの制御信号を発行することで無駄な演算の実行を停止して低消費電力化をはかる構成も可能である。

【0058】

【発明の効果】第1の効果は、IEEE-754形式を扱う浮動小数点演算器のハードウェア量を削減することができるということである。これにより、IEEE-754形式を扱う浮動小数点演算器のLSI上での実装面積が減少しマイクロプロセッサ等の集積度を向上させることができるようになる。

【0059】その理由は、従来各浮動小数点演算器で持っていたハードウェアの一部を本発明の浮動小数点演算

処理装置の外部にある記憶装置とのインタフェース部分に集約して各浮動小数点演算器から削減したからである。この効果は、スーパースカラプロセッサやVLIWプロセッサ、画像処理用の並列演算プロセッサなどの複数の浮動小数点演算器を1チップに内蔵する場合により顕著になる。

【0060】第2の効果は、IEEE-754形式を扱う浮動小数点演算器の処理時間を短縮することができるということである。これにより、IEEE-754形式を扱う浮動小数点演算器のパイプライン段数の削減やクロックサイクルの高速化が可能になりマイクロプロセッサ等の性能を向上させることができるようになる。

【0061】その理由は、浮動小数点演算器の処理と並列に処理することができないIEEE-754形式の特殊数のデータパターンに対する置き換え処理を各浮動小数点演算器から削除し、本発明の浮動小数点演算処理装置の外部にある記憶装置へと数値データ書き戻す部分に集約したからである。これは、一般にクロックサイクルの3倍程度で浮動小数点演算処理が完了するのに対して、外部の記憶装置にデータのアクセスを行う場合にはクロックサイクルの数倍から数十倍の処理時間が必要であるためこの部分でのわずかな処理時間の増大はマイクロプロセッサ全体の処理性能にほとんど影響がないために可能である。

【図面の簡単な説明】

【図1】IEEE-754形式フォーマット（倍精度）と、特定のデータの種類の表すデータパターンを説明する図である。

【図2】従来の浮動小数点演算器を説明するブロック図である。

【図3】オペランドステータスを生成するブロックの詳細を説明する図である。

【図4】浮動小数点演算器の出力結果をオペランドステータスにより差し替えるブロックの詳細を説明する図である。

【図5】本発明の一実施形態における浮動小数点演算器を説明する図である。

【図6】本発明をレジスタファイルを有するマイクロプロセッサに適用した一実施形態を示すブロック図である。

【図7】図6中のLOAD/STOREユニットの詳細を説明する図である。

【図8】図6の変形実施形態で、複数の接続経路を有する実施形態を示すブロック図である。

【図9】図6の変形実施形態で、複数の浮動小数点演算器間に相互接続経路を有する実施形態を示すブロック図である。

【符号の説明】

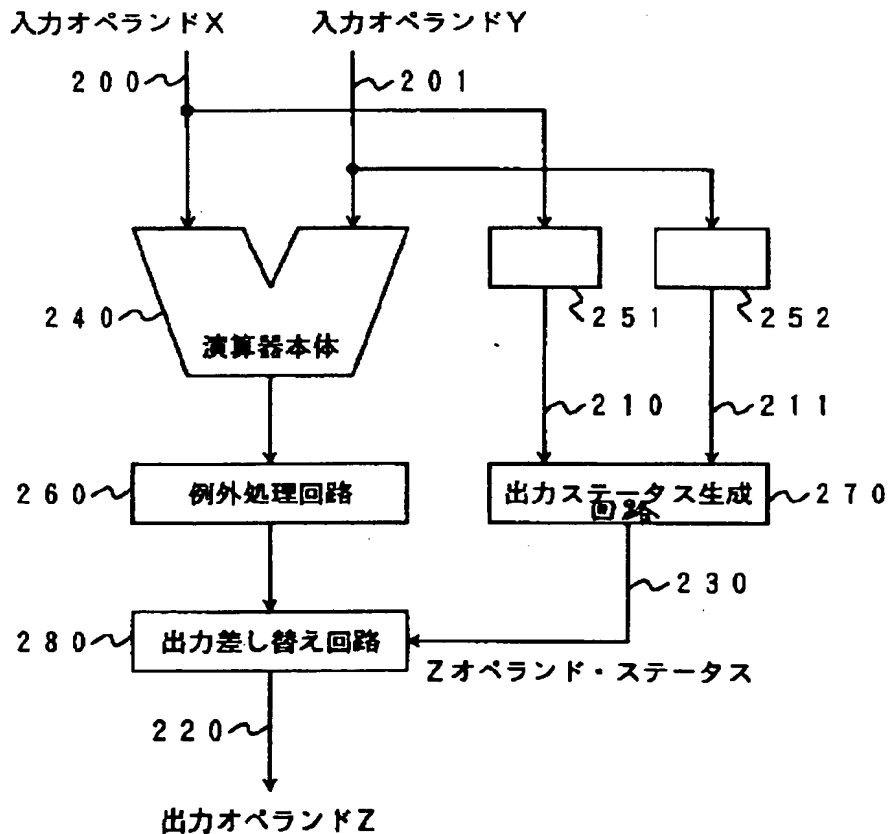
210 Xオペランドステータス信号
211 Yオペランドステータス信号

251 Xオペランドステータス生成回路
252 Yオペランドステータス生成回路
510 入力オペランドステータスX
511 入力オペランドステータスY
570 出力オペランドステータス決定論理回路
640 浮動小数点データレジスタファイル
641 浮動小数点データレジスタファイル読み出しポート1
642 浮動小数点データレジスタファイル読み出しポート2
643 浮動小数点データレジスタファイル書き込みポート
645 浮動小数点ステータスレジスタファイル
646 浮動小数点ステータスレジスタファイル読み出しポート1
647 浮動小数点ステータスレジスタファイル読み出しポート2
648 浮動小数点ステータスレジスタファイル書き込みポート
650 浮動小数点演算器本体
651 浮動小数点演算器入力オペランドデータ1
652 浮動小数点演算器入力オペランドデータ2
653 浮動小数点演算器出力オペランドデータ
655 浮動小数点演算器出力オペランドステータス決定論理回路
656 浮動小数点演算器入力オペランドステータス1
657 浮動小数点演算器入力オペランドステータス2
658 浮動小数点演算器出力オペランドステータス
660 LOAD/STOREユニット
661 STOREオペランドデータ
663 LOADオペランドデータ
666 STOREオペランドステータス
668 LOADオペランドステータス
680 メモリアクセスポート
801-1 浮動小数点オペランドデータバス1#1
801-u 浮動小数点オペランドデータバス1#u
802-1 浮動小数点オペランドデータバス2#1
802-u 浮動小数点オペランドデータバス2#u
811-1 浮動小数点オペランドステータスバス1#1
811-u 浮動小数点オペランドステータスバス1#u
812-1 浮動小数点オペランドステータスバス2#1
812-u 浮動小数点オペランドステータスバス2#u
821-1 浮動小数点オペランドデータバス3#1
821-v 浮動小数点オペランドデータバス3#v

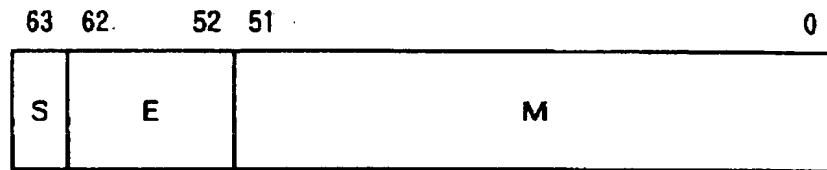
831-1 浮動小数点オペランドステータスバス
 #1
 831-v 浮動小数点オペランドステータスバス
 #v
 840 マルチポート浮動小数点データレジスタフ
 ァイル
 845 マルチポート浮動小数点ステータスレジスタ
 ファイル
 850-1 浮動小数点演算器本体#1
 850-n 浮動小数点演算器本体#n
 855-1 演算器#1用出力オペランドステータス
 決定論理回路
 855-n 演算器#n用出力オペランドステータス
 決定論理回路
 860-1 LOAD/STOREユニット#1
 860-m LOAD/STOREユニット#m
 950-1 浮動小数点演算器本体#1
 950-2 浮動小数点演算器本体#2
 953-1 浮動小数点演算器#1出力オペランドデ
 ータ
 953-2 浮動小数点演算器#2出力オペランドデ
 ータ

955-1 演算器#1用出力オペランドステータス
 決定論理回路
 955-2 演算器#2用出力オペランドステータス
 決定論理回路
 958-1 演算器#1用出力オペランドステータス
 858-2 演算器#2用出力オペランドステータス
 991-1 演算器#1用Xオペランドデータ入力選
 択回路
 991-2 演算器#2用Xオペランドデータ入力選
 択回路
 992-1 演算器#1用Yオペランドデータ入力選
 択回路
 992-2 演算器#2用Yオペランドデータ入力選
 択回路
 996-1 演算器#1用Xオペランドステータス入
 力選択回路
 996-2 演算器#2用Xオペランドステータス入
 力選択回路
 997-1 演算器#1用Yオペランドステータス入
 力選択回路
 997-2 演算器#2用Yオペランドステータス入
 力選択回路

【図2】



【図1】



S: 符号 (1 b i t)

E: 指数 (1 0 b i t)

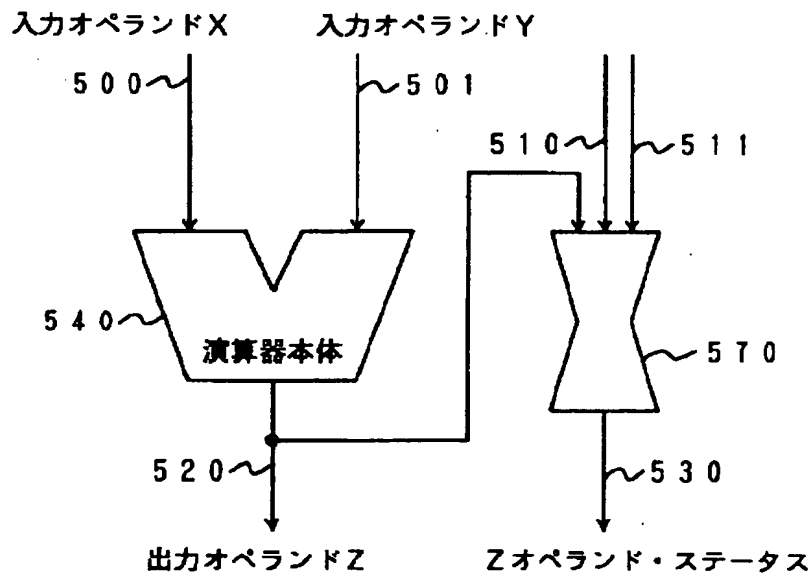
M: 仮数 (5 2 b i t)

数の種類	指数の値	仮数の値	数値例
Z E R O	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 h
∞	2 0 4 7	0	7 F F 0 0 0 0 0 0 0 0 0 0 0 0 h
s N a N	2 0 4 7	0 以外*1	7 F F F F F F F F F F F F F F F F h
d N a N	2 0 4 7	0 以外*2	7 F F 7 F F F F F F F F F F F F h
denormal	0	0 以外	0 0 0 F F F F F F F F F F F F F F h
normal	1 ~ 2 0 4 6	問わない	

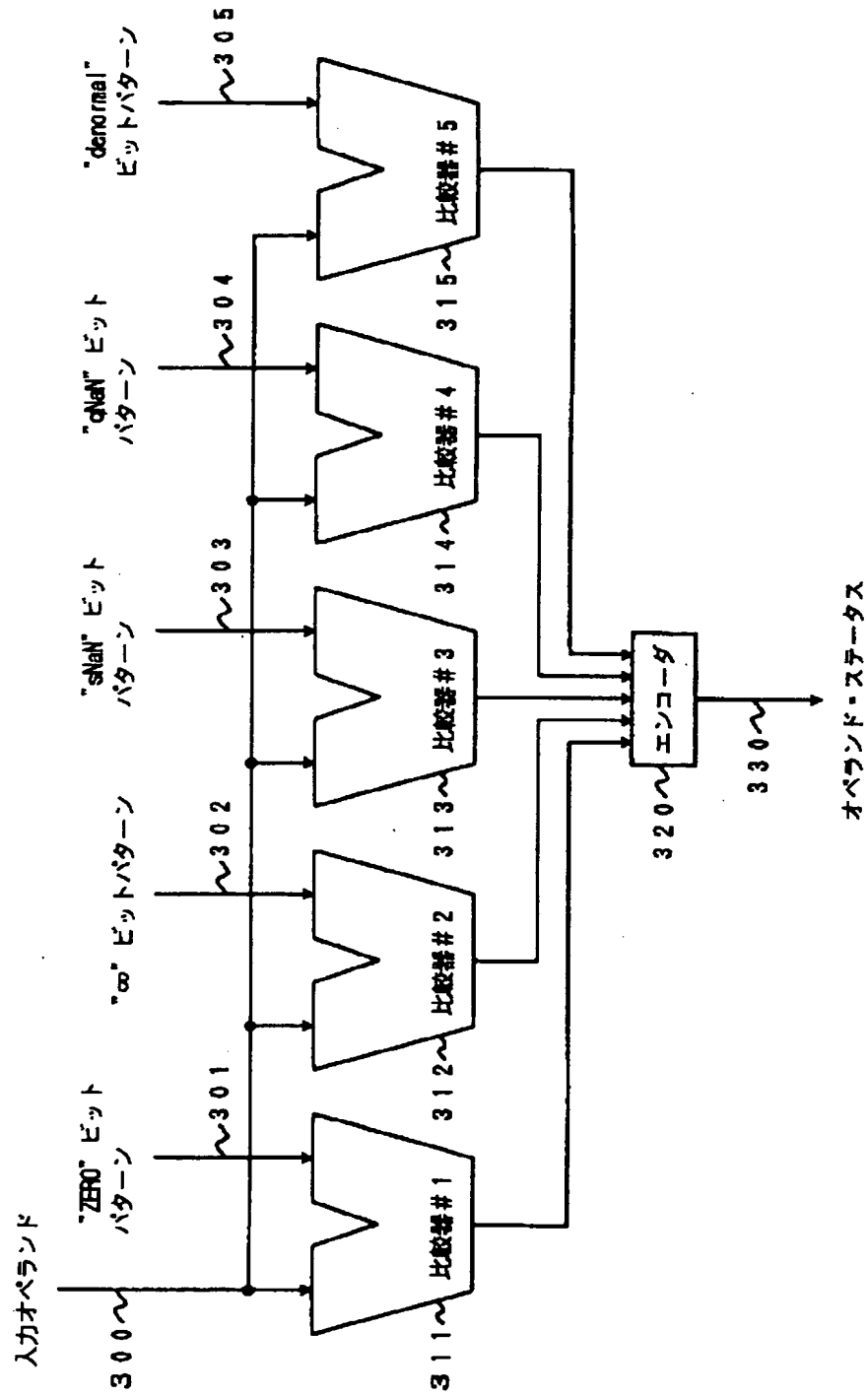
* 1 仮数の先頭 1 b i t が 1 であること

* 2 仮数の先頭 1 b i t が 0 であること

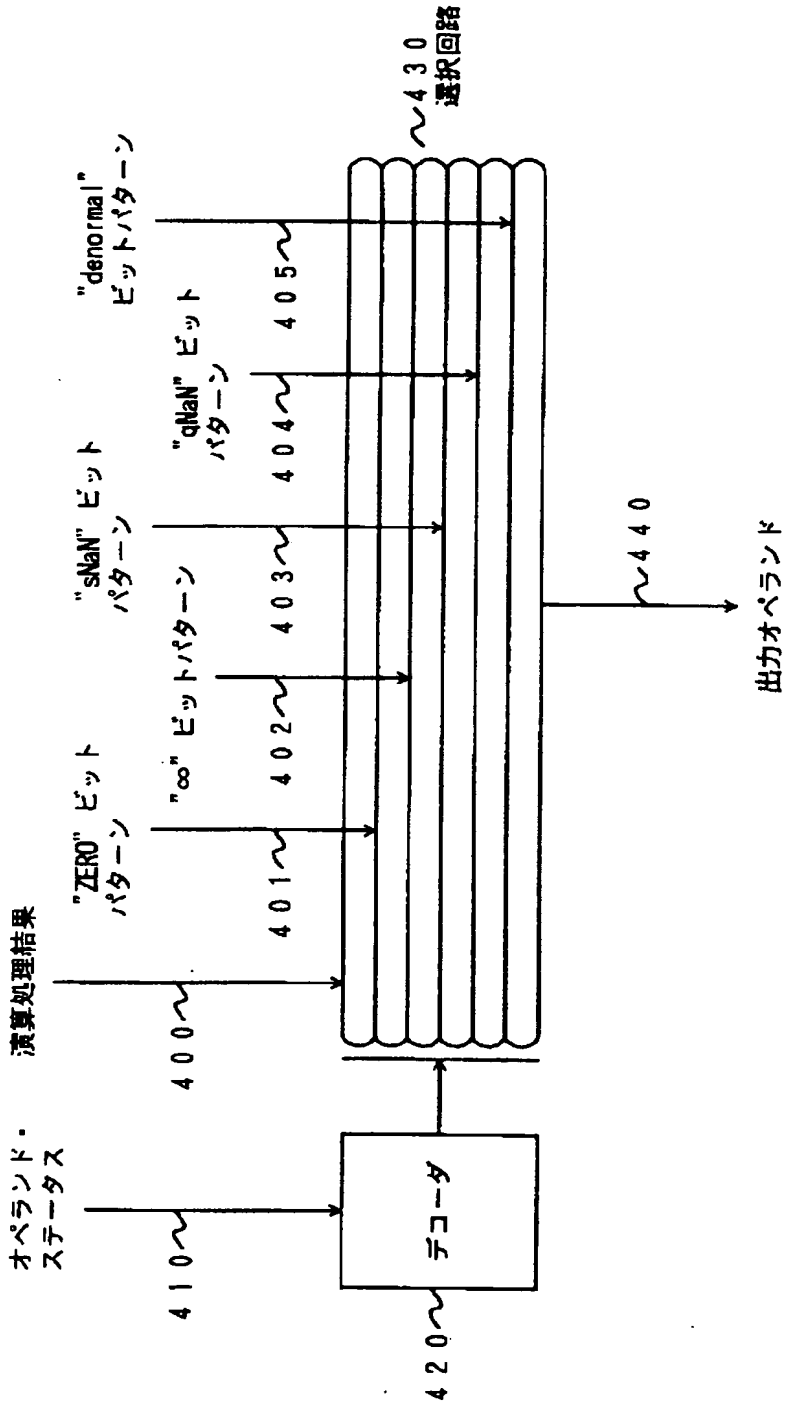
【図5】



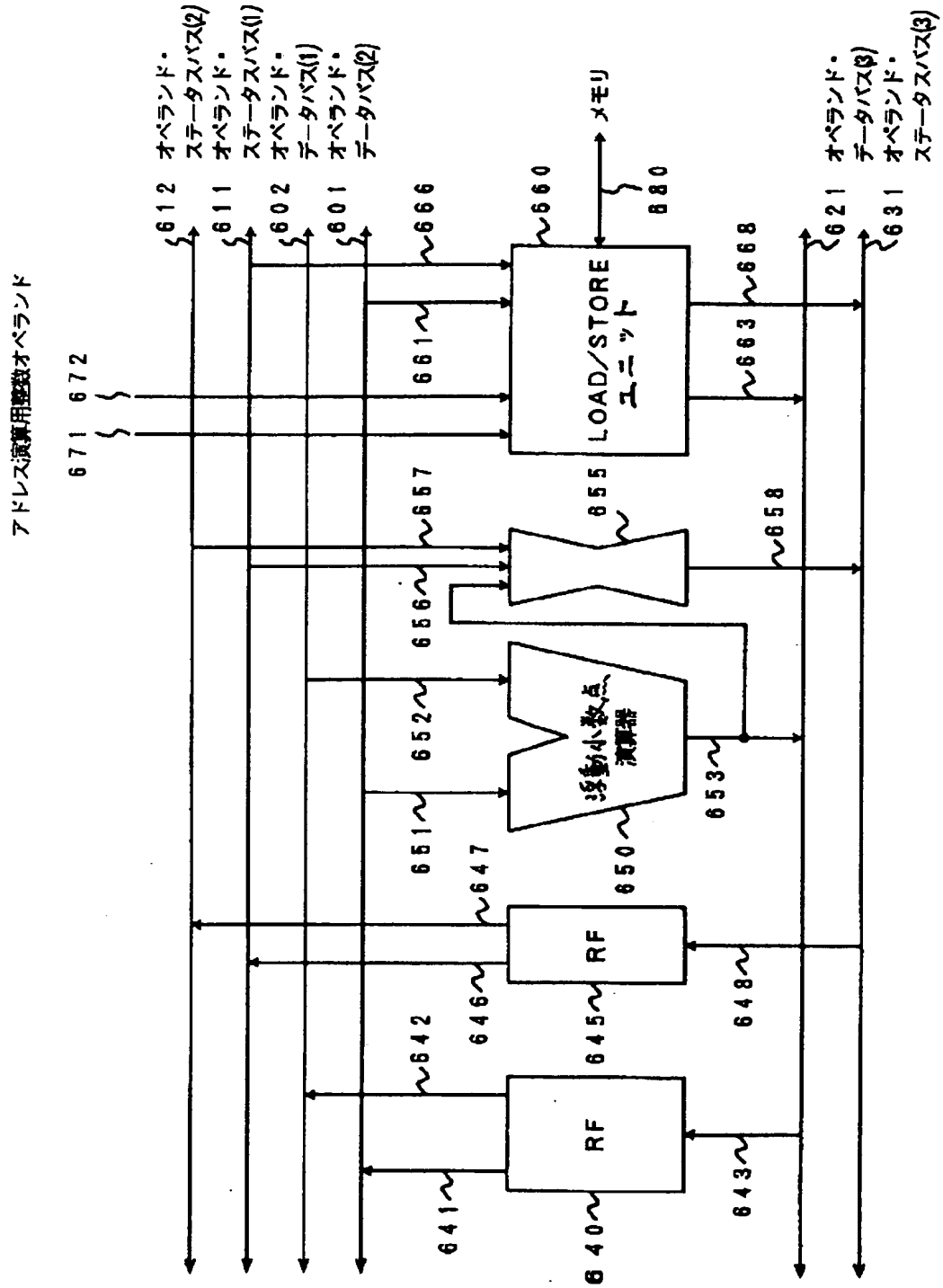
【図3】



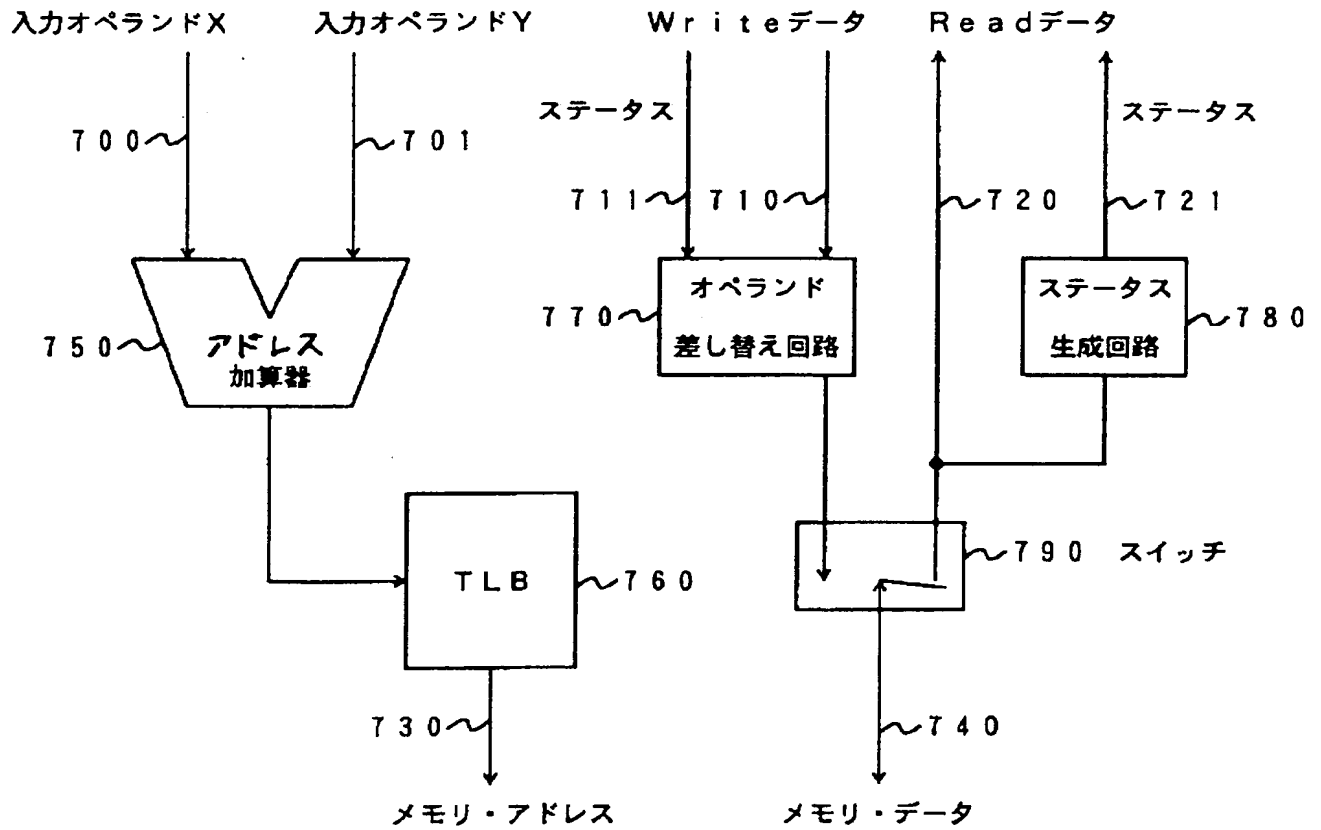
【図4】



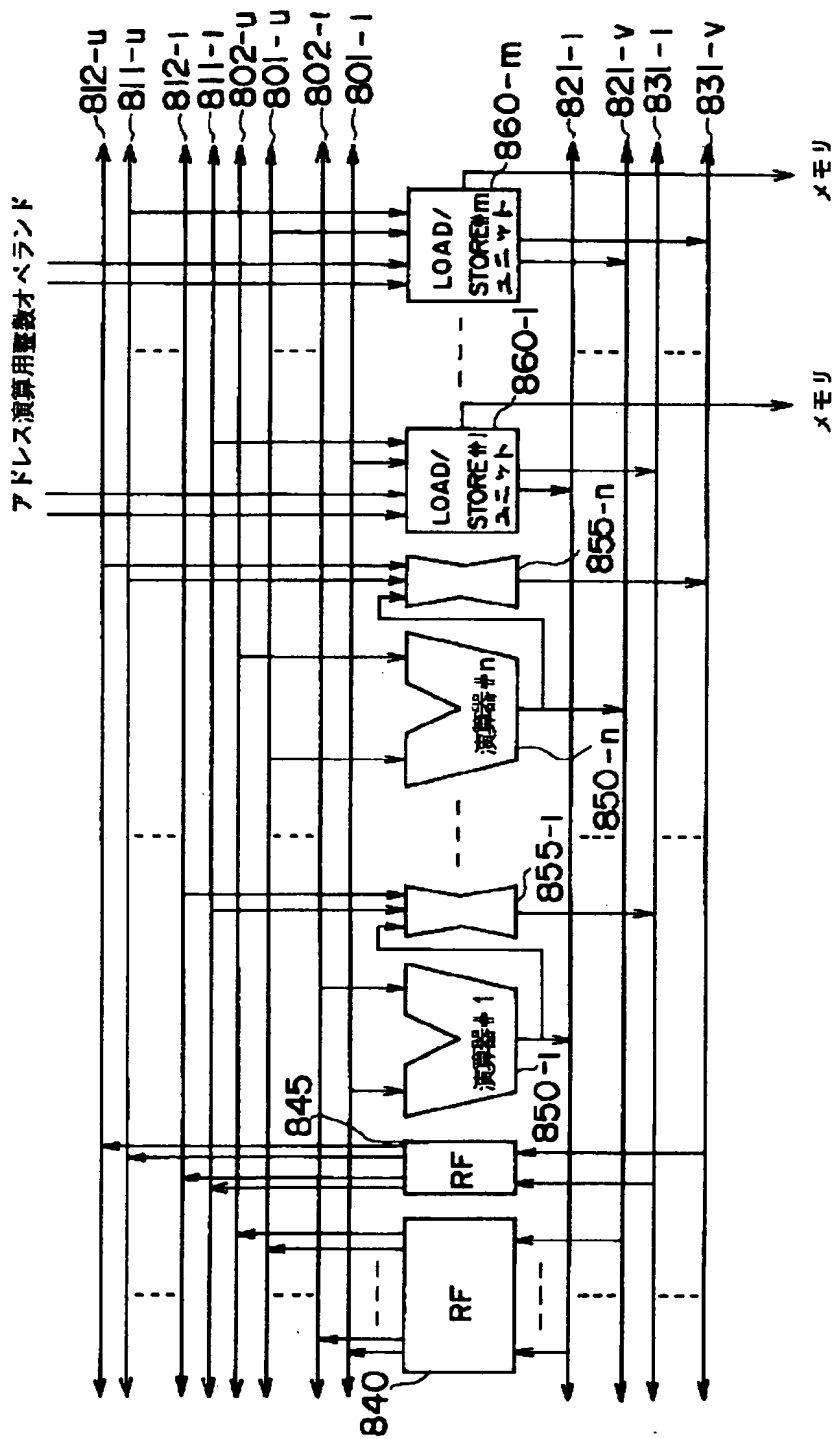
【図6】



【図7】



【図8】



【図9】

